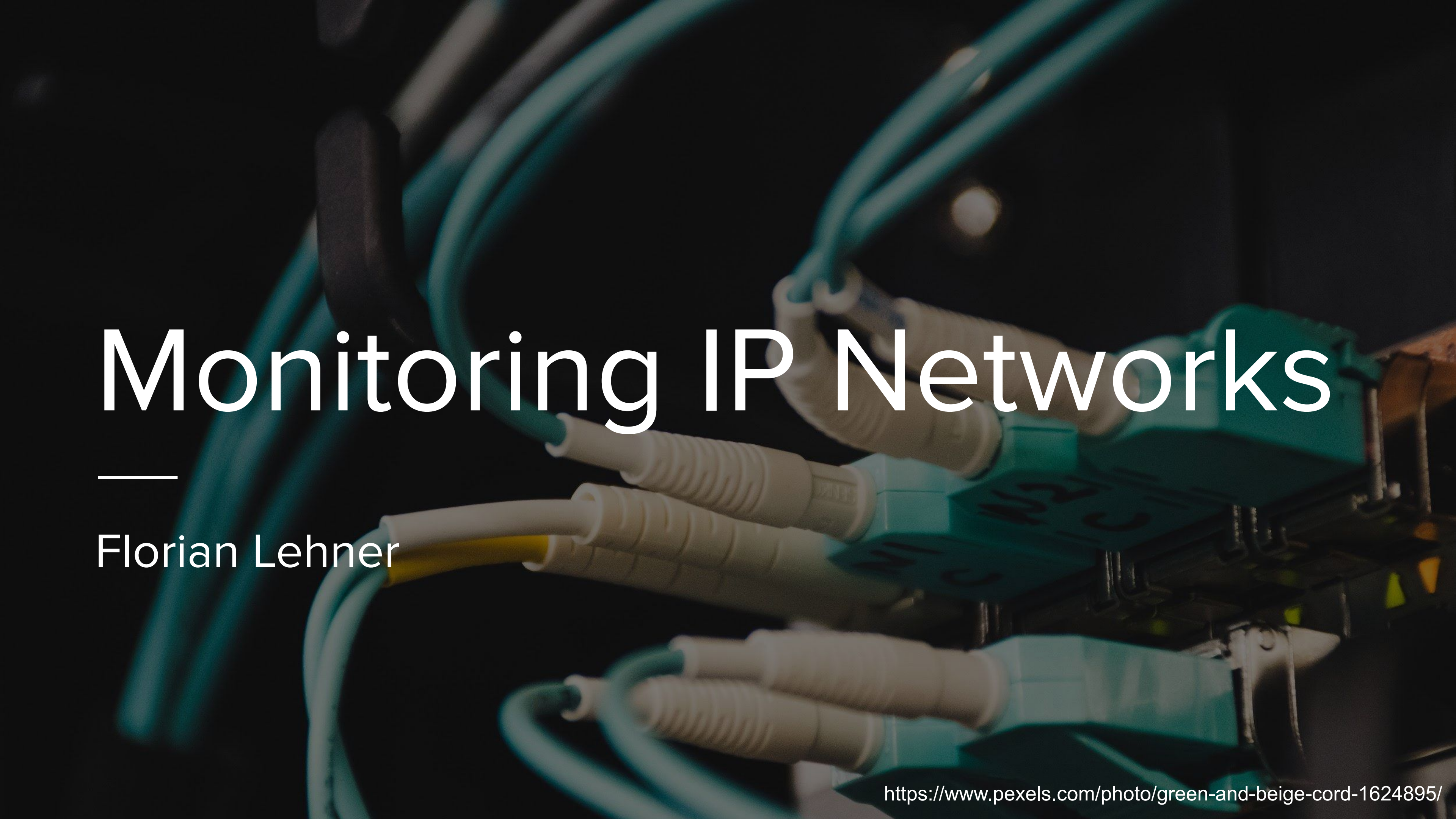


Monitoring IP Networks

Florian Lehner

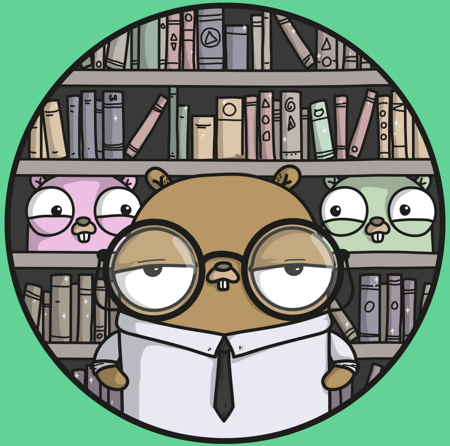
https://www.pexels.com/photo/green-and-beige-cord-1624895/

Florian Lehner

- Network Security Engineer at Open Systems*
- Member of AS 57821
- github.com/florianl/

* **Disclaimer**: This talk is about private projects and **not related** to Open Systems.

Observability is an essential part of operating an IP network



What to monitor?

- interface
- source and destination IP address
- IP protocol
- length of IP packet

Agenda

1. tcpdump
2. netfilter
3. traffic control

tcpdump

- cgo*/unsafe
- huge variety of decoders
- huge memory consuming footprint
- information is packet based

```
import "github.com/google/gopacket"

// Prepare decoders for expected layers
parser := gopacket.NewDecodingLayerParser(...)

// Attach the capture to the given interface
handle, err := pcap.OpenLive(...)

// Create a packet data source
packetSource := gopacket.NewPacketSource(...)

for packetData := range packetSource.Packets() {
	// Try to decode each received packet
	// and extract its information
	parser.DecodeLayers(...)
	for _, layerType := range decoded {
		switch layerType {
			[...]
		}
	}
}
```

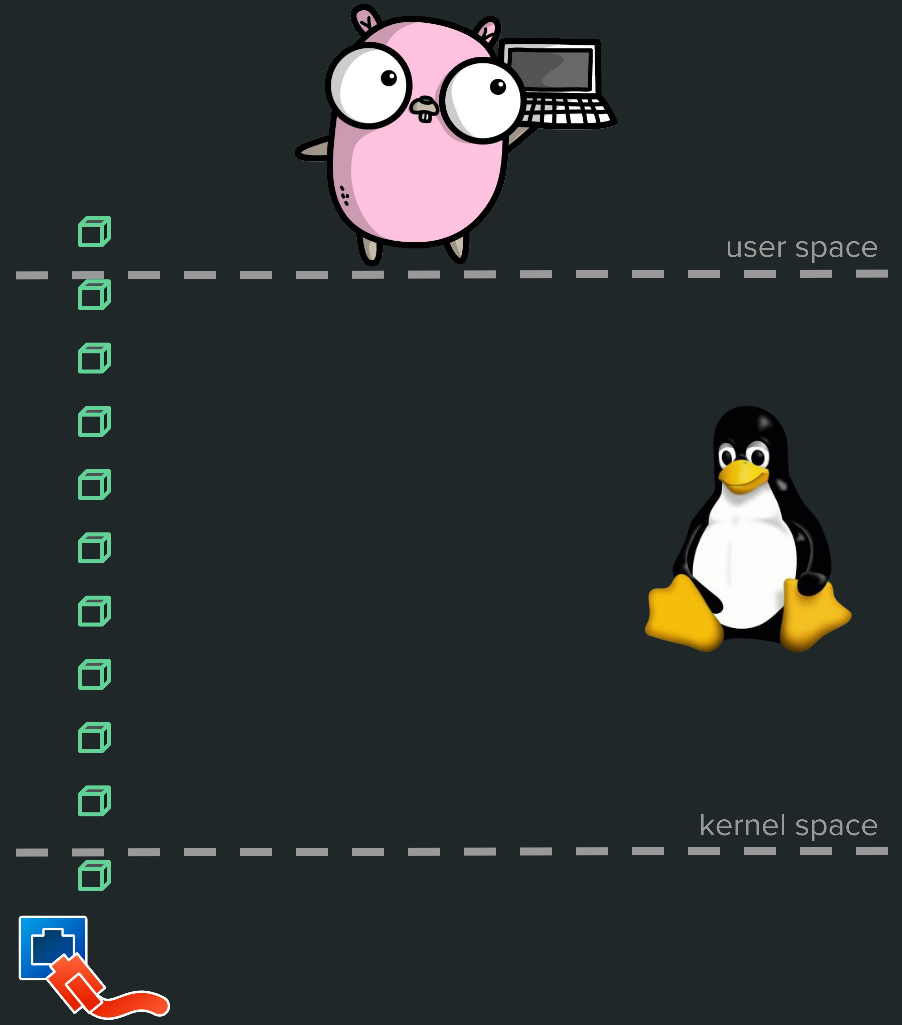
* github.com/notti/nocgo

https://github.com/florianl/monitoringIPbasedNetworks/blob/master/libpcap/main.go

tcpdump

- cgo*/unsafe
- huge variety of decoders
- huge memory consuming footprint
- information is packet based

is there a better way?



* github.com/notti/nocgo

netfilter log

- needs special firewall rules
- information is packet based
- parsing still needs to be done

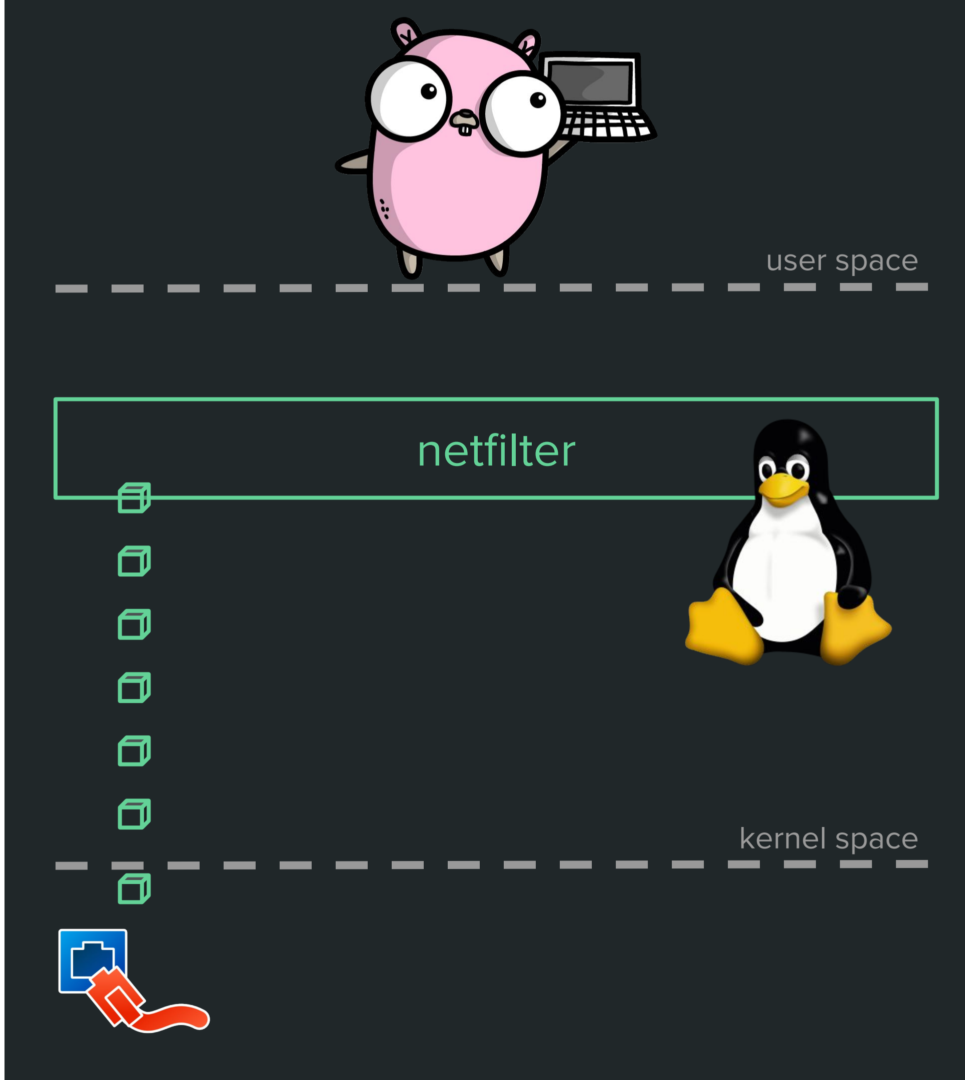
```
// Send all incoming traffic to nflog group 100
// sudo iptables -I INPUT -j NFLOG --nflog-group 100

// Open netlink socket for nfnetlink_log
nf, err := nflog.Open(...)

// Define a hook to handle received packets
fn := func(m nflog.Msg) int {
      [...]
}

// Register the hook on the nfnetlink_log socket
nf.Register(fn, ...)
```

https://github.com/florianl/monitoringIPbasedNetworks/blob/master/libnetfilter_log/main.go

netfilter log

- needs special firewall rules
- information is packet based
- parsing still needs to be done

is there a better way?



netfilter conntrack

- information is session based
- not all needed information is included (interface is missing)
- ENOBUF

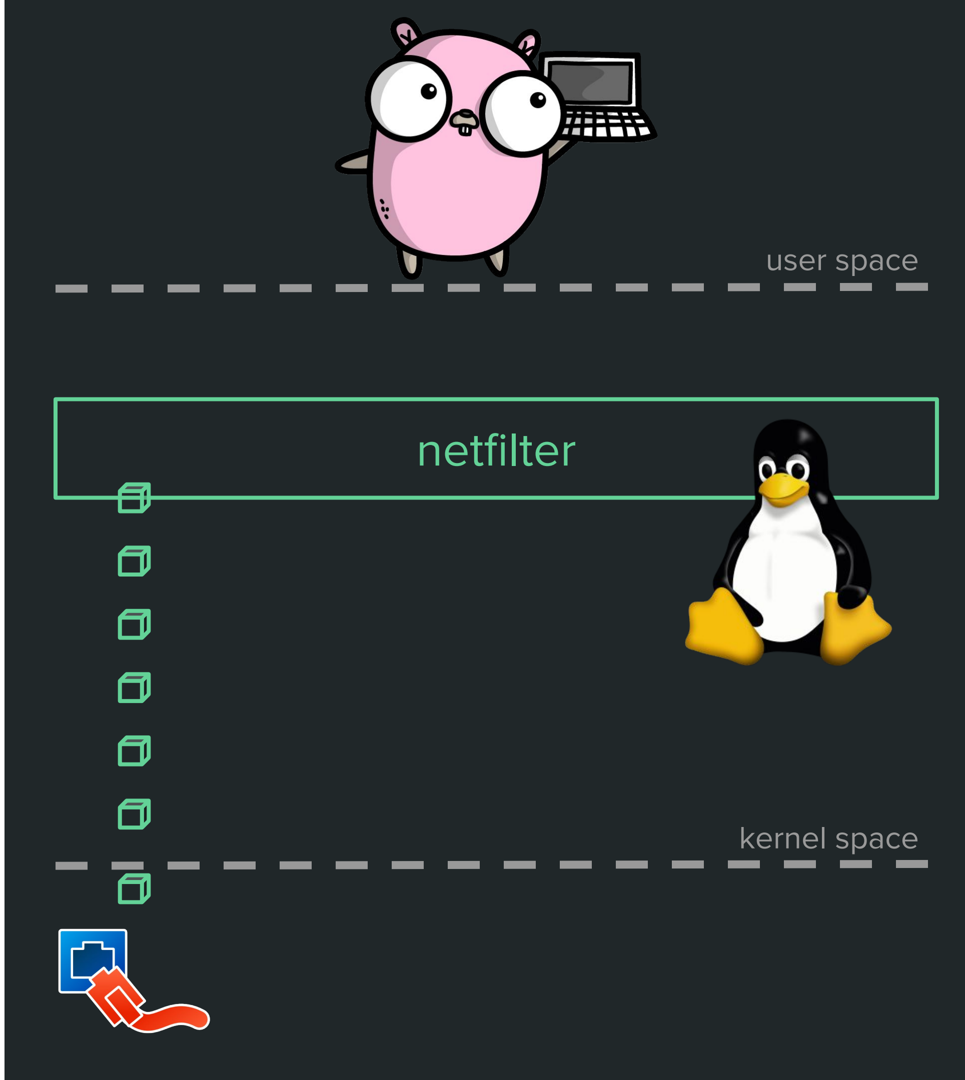
```
// Open netlink socket for nfnetlink_conntrack
nfct, err := ct.Open(...)

// Define a hook to handle received packets
fn := func(c ct.Conn) int {
        [...]
}

// Register the hook for New|Update|Destroy events
nfct.Register(fn, ...)
```

https://github.com/florianl/monitoringIPbasedNetworks/blob/master/libnetfilter_conntrack/main.go

netfilter conntrack

- information is session based
- not all needed information is included (interface is missing)
- ENOBUF

is there a better way?



one cannot talk about observability without mentioning eBPF

What is eBPF?

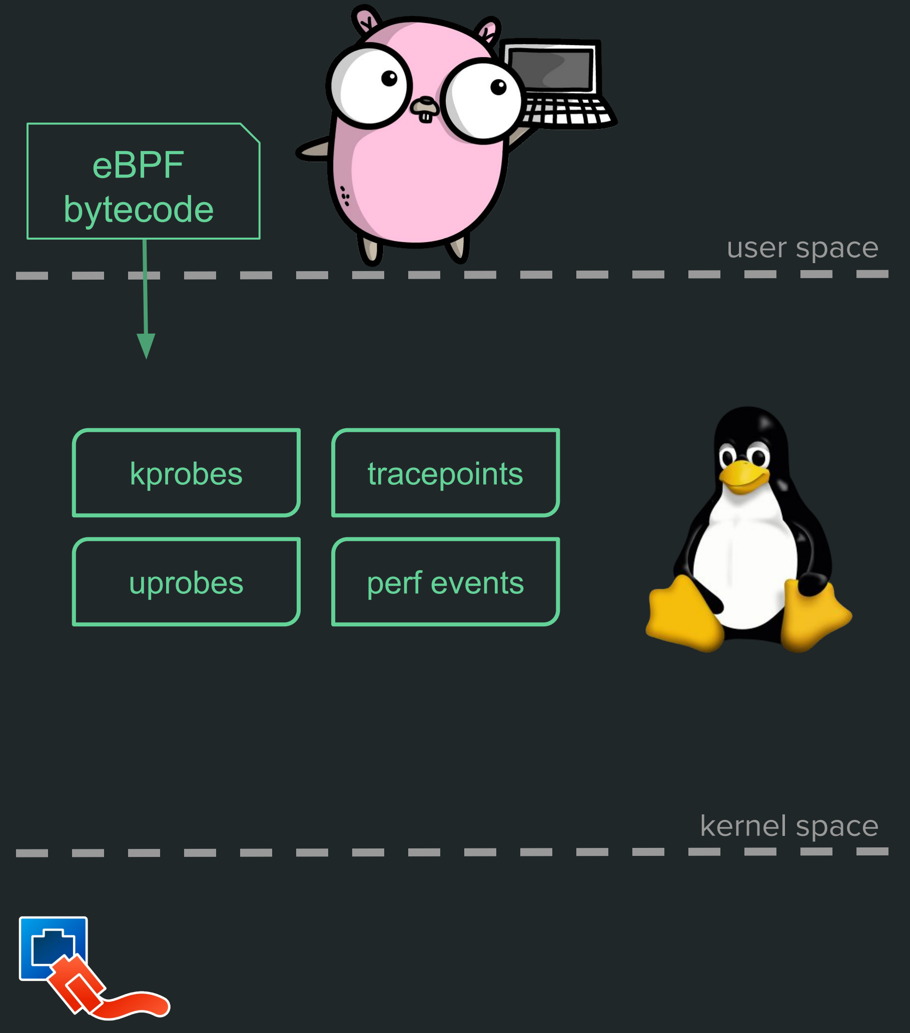
"What Javascript is to HTML, BPF is to the Linux kernel"

– Beatriz Martínez Rubio (IBM) @ KubeCon 2019

"crazy stuff"

– Alexei Starovoitov (eBPF lead)

http://man7.org/linux/man-pages/man2/bpf.2.html
http://www.brendangregg.com/ebpf.html



traffic control

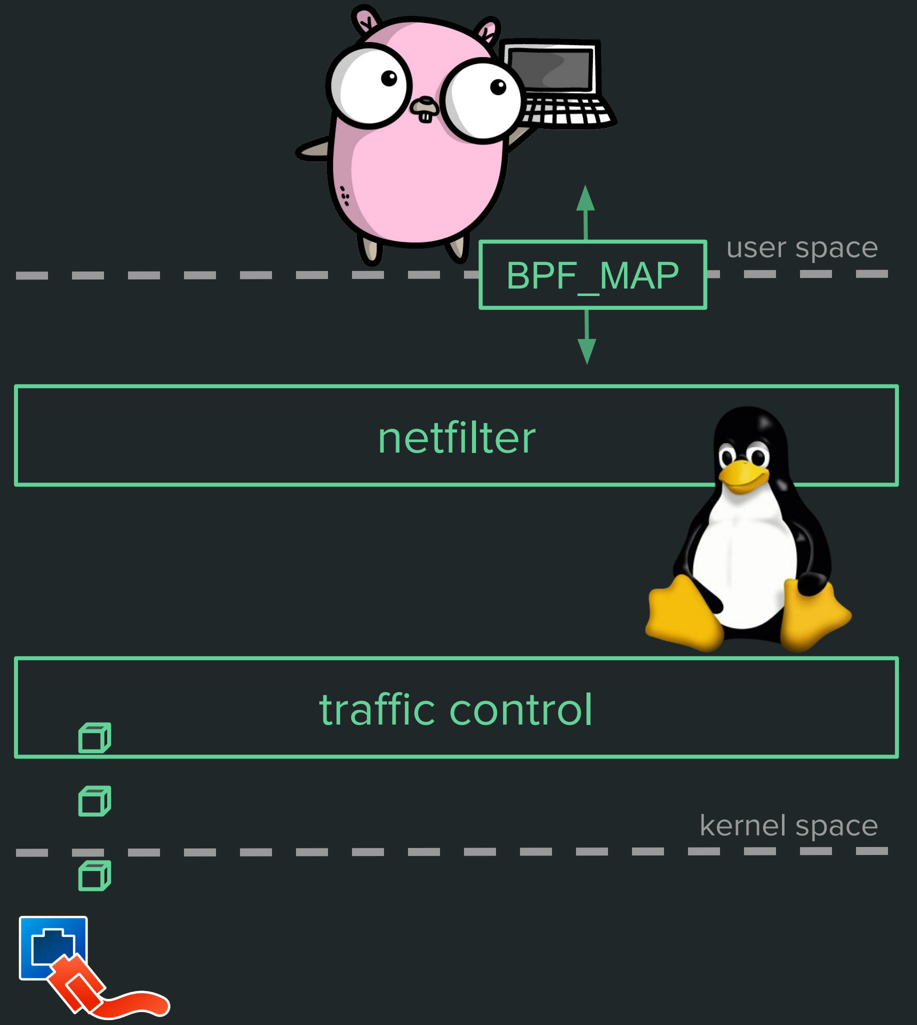
- kernel code in C
- use of in kernel structs
- extract only needed data

```
import "C"

// Write your eBPF module in C
const source string = `
#include <uapi/linux/bpf.h>

int tcExample(...) {
      ...
}
`

// Create an eBPF module
module := bpf.NewModule(...)

// Open a netlink socket
rtnl, err := tc.Open(...)

// Add a queueing discipline
rtnl.Qdisc().Add(...)

// Add filter with the eBPF module
rtnl.Filter().Add(...)

for {
      // handle data from the eBPF module
      data := <-channel
}
```

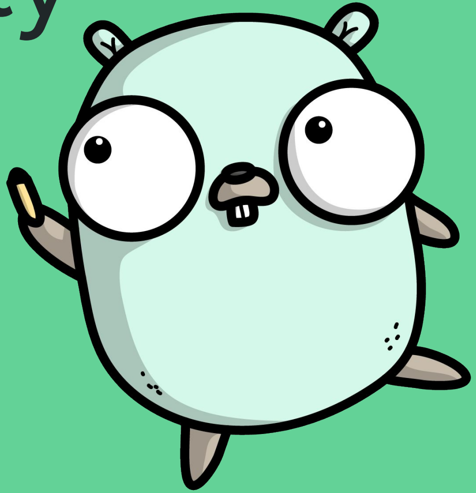
https://github.com/florianl/monitoringIPbasedNetworks/blob/master/eBPF/trafficControl/main.go

traffic control

- kernel code in C
- use of in kernel structs
- extract only needed data



Go offers various ways to improve observability in your IP network.



Conclusion

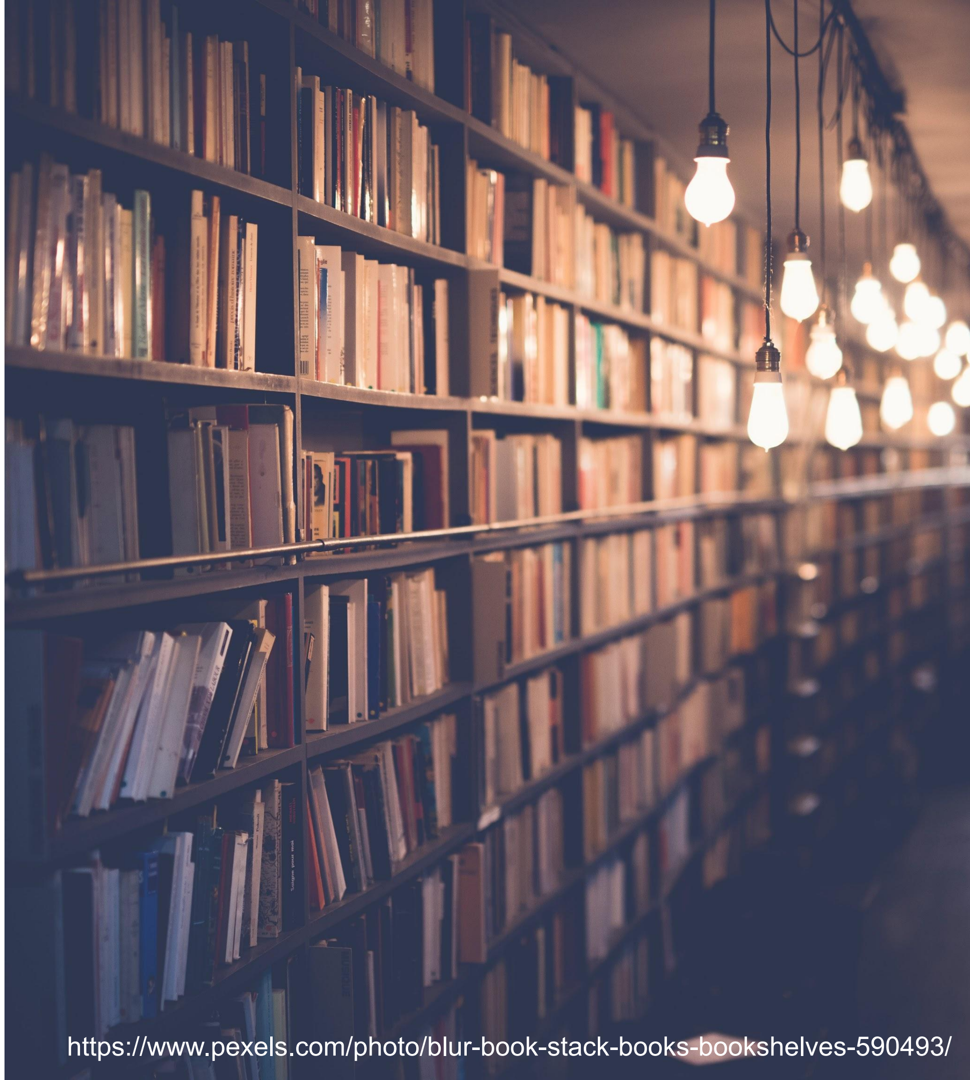
Go allows low level observability of IP traffic

Questions?

full code examples
github.com/florianl/monitoringIPbasedNetworks

Gophers by
github.com/ashleymcnamara/gophers

Tux by
wikipedia.org/wiki/Tux_(mascot)


https://www.pexels.com/photo/blur-book-stack-books-bookshelves-590493/